A Multi-Algorithm, High Reliability, Extensible Steganalyzer using Services Oriented Architecture

Eman Abdelfattah
University of Bridgeport
221 University Avenue
Bridgeport, CT 06604

Ausif Mahmood
University of Bridgeport
221 University Avenue
Bridgeport, CT 06604

ABSTRACT

This paper presents a unified Steganalyzer that can work with different media types such as images and audios. It is also capable of providing improved accuracy in stego detection through the use of multiple algorithms. The designed system integrates different steganalysis techniques in a reliable Steganalyzer by using a Services Oriented Architecture (SOA). Other contributions of the research done in this paper include, an improved Mel-Cepstrum technique for audio *wav* files feature extraction that results in better accuracy in stego detection (> 99.9%), improved overall classification system that is based on three individual classifiers; a Neural Network classifier, a Support Vector Machines classifier, and an AdaBoost algorithm based classifier. Finally, an extensible classifier is introduced that allows incorporation of detecting new embedding techniques to the current system, so that the framework will continue to provide reliable stego detection for future embedding algorithms.

General Terms

Classification, Steganalysis, Services Oriented Architecture

Keywords

Mel-Cepstrum, Support Vector Machines, Neural Networks, AdaBoost

1. INTRODUCTION

Network security has received increased attention in the last decades. Encryption has laid itself as the traditional method to transmit information in secrecy. Although strong encryption is a very secure approach for transmitting information, it can be easily identified that transmitted information is encrypted. Once the information is identified as encrypted, an intruder can block the encrypted transmission. In contrast, steganography is a viable option to hide information in transmission without being identified. It provides a blanket that hides encrypted information. Thus for security purposes, it becomes essential to develop mechanisms that reveal if the communicated information has any embedded data. These mechanisms are labeled as steganalysis. Fridrich defines steganalysis as “the art of discovering hidden data in cover objects.” [1]. The cover object is an object of any type such as an image or an audio file that contains no hidden information. A stego object is obtained by modifying the cover object using an embedding algorithm.

The main focus in steganalysis is to detect the presence of a hidden message in a stego object. The steganalysis techniques are classified under two categories; specific and universal steganalysis. The specific steganalysis techniques are designed for a targeted embedding technique. Thus, they yield very accurate decisions when they are used against a particular steganographic technique. In universal techniques, dependency on the behavior of the individual embedding techniques is removed by determining and collecting a set of distinguishing statistics that are sensitive to a wide variety of embedding operations. Universal steganalysis techniques are composed of two important components; feature extraction, and feature classification. In feature extraction, a set of distinguishing statistics are obtained from a data set of objects by observing general object features that exhibit a strong variation under embedding. The feature classification uses distinguishing statistics from both cover and stego objects to train a classifier. Then, the trained classifier is used to classify an input object as either cover or stego. Since different carrier types can be used and there are many possible embedding techniques, it makes designing a reliable steganalyzer very challenging. We present a universal steganalyzer that not only works with different carrier types, but also develops an extensible multi-algorithm framework based on SOA that allows for easy incorporation of new carriers and embedding algorithms.

The remaining paper is organized as follows. In section 2, we present a literature survey of some existing steganography and steganalysis techniques. In section 3, we describe our new SOA based framework for the universal steganalyzer. In section 4, we present Mel-Cepstrum based steganalysis technique used in audio files. Then, in section 5, we report results and analysis. Finally, we offer conclusions in section 6.

2. EXISTING WORK IN STEGANALYSIS

We divide the steganography and steganalysis techniques into three main categories depending upon the carrier being used. These include techniques based on images, based on multimedia (e.g., audio), and using other carriers such as HTML files or torrent files. We present the existing state of research in these categories in the following subsections.

2.1 Image Steganography and Steganalysis

With the wide availability of digital images, and the high degree of redundancy present in them despite compression, there has been an increased interest in using digital images as cover-objects for the purpose of steganography [2]. Martín *et. al.* investigate the effect of embedding on the statistics of the image to detect the presence of a secret message [3]. The three different stego algorithms that are used in their work are: Jsteg [4], MHPDM [5], and one of the algorithms in S-Tools [6]. The following five different statistical models of natural images are used: Areas of Connected Components Model [7, 8], Adjacent Pixel Values Model [9-11] Laplacian Distribution Model [12], Wavelet Coefficients Model and DCT Coefficients Model.

Martín *et. al* conclude that the effect of the embedding operations is insignificant to the natural images when the analysis is independent of the steganograhy algorithms.

Avcibas *et. al.* use binary Similarity Measures (BSMs) to calculate three types of features; computed similarity differences, histogram and entropy related features, and a set of measures based on a neighborhood-weighting mask by looking at the seventh and the eighth bit planes of an image [13, 14]. The authors conclude that their technique demonstrates comparable results to the results obtained by Farid's scheme [15] which uses a wavelet based decomposition to build higher-order statistical models.

Kharrazi *et. al.* study the performance of three distinct blind steganalysis techniques against four different steganographic embedding techniques. The three steganalysis techniques used for feature extraction are binary similarity measures (BSMs), wavelet-based steganalysis (WBS) and feature-based steganalysis (FBS). The analyzed four embedding techniques are: Outguess [16], F5 [17], Model-Based [18] and perturbed quantization PQ [19]. The used cover media is *jpeg* images. Kharrazi *et. al.* conclude that FBS achieves superior performance because the used data set is compressed *jpeg* images [20]. Moreover, PQ steganography embedding technique is the best one because it is the least detectable technique. Other works in the area of image steganalysis are found in [1, 21-24].

2.2 Audio Steganography and Steganalysis

Tian *et. al.* propose an m-sequence based Steganography technique for Voice over IP [25]. The technique succeeds to achieve good security, sufficient capacity and low latency by using least-significant-bits (LSB) substitution method. Moreover, m-sequence encryption approach is used to eliminate the correlation among secret messages so that the statistical steganalysis algorithm can hardly detect stego-speech.

Liu *et. al.* present two methods. In the first method, the statistics of the high-frequency spectrum and the Mel-Cepstrum coefficients of the second-order derivative are extracted for audio steganalysis. In the second method, a wavelet-based spectrum and Mel-Cepstrum coefficients are extracted for audio steganalysis [26]. A comparison among these two methods and the signal-based Mel-Cepstrum audio steganalysis method is conducted. Liu *et. al.* conclude that the proposed methods outperform the signal-based Mel-Cepstrum approach. Moreover, the derivative-based approach outperforms the wavelet-based approach. Other works in the area of audio steganalysis are found in [27-29].

2.3 Other Media used in Steganography and Steganalysis

In addition to the previously discussed carriers, some other digital entities can be used as cover media. For example, HTML files (hypertext markup language) have appropriate potentials for information hiding. While processing HTML files, the browser ignores spaces, tabs, certain characters and extra line breaks which could be used as locations for hiding information. Another example of carrier media is the unused or reserved space on a disk to hide the information. Also, data can be hidden in unused space in file headers. Furthermore, network protocols such as TCP, UDP, and/or IP can be used for hiding the messages and transmit them through the network [30]. Li *et. al.* suggest using torrent files, a crucial part of the BitTorrent P2P network, as host carriers for secret messages [31].

Since there are numerous embedding techniques (which may continue to increase) and many possible carrier types, we propose to develop an SOA based Steganalyzer that is extensible to not only different carrier types but also different embedding algorithms. We present our unified steganalysis framework in the next section.

3. A NEW FRAMEWORK FOR STEGANALYSIS

We introduce a new framework for a reliable and extensible steganalyzer using Services Oriented Architecture (SOA) that integrates multiple algorithms and can handle different carrier types. In our SOA design, the system is broken down into independent services. The choice of SOA is motivated by the fact that a reliable steganalyzer needs to incorporate many different algorithms and that it should continue to evolve as newer embedding and stego detection techniques are developed. SOA also presents advantages such as "high flexibility, simplicity, maintainability, and reusability. Furthermore, it provides platform independency and distributes the overall load of the process" [32].

The motivation behind using an SOA in a software implementation is its flexibility in providing an independent execution environment from the service interface. SOA architectures were proposed to provide an integrated solution for many obstacles that developers face in a distributed enterprise computing environment. These obstacles include application integration, security issues across multiple platforms and protocols, and transaction management. SOA aims to diminish these obstacles so that applications across different platforms and systems run seamlessly [33]. Detailed definitions of SOA can be found at [34-36].

The advantages of the new framework are:

a. A reliable implementation of the steganalysis complex problem.
b. Extensibility of the system where more services for new steganalysis techniques can be added later.
c. Flexibility, where a better service can replace an existing service when an improved steganalysis technique is developed for this specific category.

The proposed architecture includes different services as follows:

1. An extension service that can identify different types of carriers such as images and audios.
2. Services that handle different steganalysis techniques.

Our new SOA based Steganalyzer contains the components shown in Figure 1. We present an enhancement of this initial framework later in this paper. The extension service is the component responsible for identifying the type of files under steganalysis. The complexity service is implemented for the files that have an extension of *.wav*. The choice of two different service implementations of *.wav* files is motivated by the reported results in [28, 37] that as the complexity of the *.wav* audio files increases, a different technique can be used to obtain better stego detection accuracy results. The Mel-Cepstrum service provided by our steganalyzer is based on Mel-frequency cepstrum technique in conjunction with second order derivative, which is widely used in image processing to detect isolated points, edges, etc [38]. A Markov service based on the second order derivative of audio signals suggested by Liu *et. al.* is implemented [28]. In our steganalyzer, we implement this technique for extracting the features in case the complexity of the audio signal is greater than or equals to 0.08 to obtain a better accuracy as suggested by Liu *et. al.* [28]. The intra-Blocks service is used to handle *jpeg* images. The feature extraction technique that is used in this service is based on the work of *Liu et. al.* [39]. The neighboring joint density features on intra-blocks are extracted. The details of the architectural components are given in [40].

4. IMPROVED MEL-CEPSTRUM BASED STEGANALYSIS

We have introduced a feature extraction technique based on Mel-frequency cepstrum in conjunction with second order derivative in [40]. In this section, we present an improved Mel-frequency cepstrum approach.

Inspired by Liu *et. al.* technique [28], we extract the Mel-Cepstrum and filtered Mel-Cepstrum features using the following equations:

$$MelCepstrum = FT\left(MT\left(FT\left(D_f^2\right)\right)\right) = \begin{pmatrix} sf_{mel1} \\ sf_{mel2} \\ \cdots \\ sf_{melC} \end{pmatrix} \qquad (1)$$

$$\begin{aligned} &FilteredMelCepstrum \\ &= FT\left(SpeachBandFiltering(MT\left(FT\left(D_f^2\right)\right)\right)\right) \\ &= \begin{pmatrix} sf_{mel1} \\ sf_{mel2} \\ \cdots \\ sf_{melC} \end{pmatrix} \qquad (2) \end{aligned}$$

In reference to equation (2), a butterworth band-stop filter is used to remove the spectrum components between 200 and 6810.59 Hz. A total of 58 features are extracted, 29 MFCCs features and 29 FMFCCs features. The code used to calculate these features is based on the code available at [41]. The following methodology is employed:

(1) A signal of duration *T* is divided into windows. A total number of frames can be calculated as follows:

Total number of frames = T / S + 1

Where S is the successive period between windows.

(2) A total of 29 MFCCs features, MFCC[1] to MFCC [29], and 29 FMFCCs features, FMFCC[1] to FMFCC[29], are extracted per audio frame.

(3) A total number of 58 * *(T / S + 1)* features are extracted.

(4) The MFCC[1] coefficient in each of the *(T / S + 1)* frames represents the signal energy.

(5) The extracted features of the cover audios and stego audios are analyzed to study the effect of embedding on different frames of the stego audios to select good features that can be used to distinguish between the cover and the stego file.

4.1 Classifiers Used in Our Framework

In this section, we briefly discuss three different classifiers used for classifying an object as stego or cover. The three classifiers used in our work are: Support Vector Machines, AdaBoost and Neural Networks.

Support Vector Machines (SVMs) are used extensively in literature for different pattern recognition and classification problems. A SVM is a heuristic approach that had shown competitive results to other heuristic approaches such as neural networks and fuzzy logic in different pattern classification problems. SVMs are one of the classifiers that are used in the problem of steganalysis in this paper.

AdaBoost is a methodology that was introduced in 1995 by Yoav Freund and Robert Shapire [42] to enhance the classification results obtained from a set of weak classifiers. The details of this methodology can be found in [43].

A Neural Network (NN) is a predictive model that aims to map a set of input patterns onto a set of output patterns. A series of input/output data sets is used to train the network. Then, the trained network applies what it has learned to predict the corresponding output [44]. Neural Networks have proven to be effective in accomplishing good results in classification problems. Using a neural network involves two processes; training and testing. In our implementation, in the training process, a feed-forward backpropagation neural network is created. The Matlab differentiable transfer function *tansig* is employed. The backpropagation training function *tainrp* is selected for its memory utilization. A neural network of 4-layers or 5-layers is constructed in case the number of features is 58 or 169 respectively as shown in Figure 1. More details are explained in [40].

4.2 Testing Environment

The initial Services Oriented Architecture framework developed in this work is implemented using Windows Server 2008 platform and WCF services (based on WS-* SOA standards) for SOA implementation. We have used the following methodology in implementing the framework:

(A) The feature extraction algorithms for audio or image files using one of the techniques given in Section 3 are implemented in Matlab.

(B) Classifiers using SVMs are constructed in Matlab.

(C) A total of 50 experiments are conducted using SVMs. In each experiment, 50% of the files are randomly used for training and 50% are assigned for testing.

(D) Classifiers using N-layer feed-forward backpropagation neural networks are constructed in Matlab.

(E) Each of the neural networks is constructed and percentage of the data set of the cover and the stego features are randomly selected to train the network.

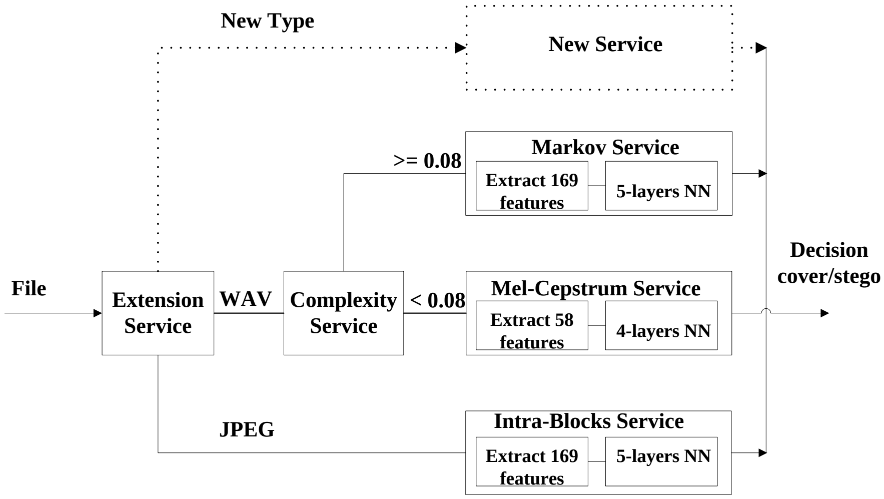


Fig 1: An Overview of the Initial Services Oriented Architecture based Steganalyzer

(F) The remaining data set of the cover and the stego features are used to test the trained network to determine each of the neural network's accuracy.

(G) The feature extraction code for Mel-Cepstrum, Markov and intra-Blocks techniques and the corresponding trained neural network classifiers are converted (using deployment tool in Matlab) to dynamic link libraries (DLLs) to be integrated in Mel-Cepstrum service, Markov service and intra-Blocks service respectively. Then, dynamic link libraries are integrated into the web services in our SOA architecture.

5. RESULTS AND ANALYSIS

In this section, we report and analyze the results of implementing two different steganalysis techniques. These techniques are: Intra-Blocks technique for *jpeg* images and improved Mel-Cepstrum technique for *wav* audios.

5.1 Intra-Blocks Technique

A total of 5151 *jpeg* images are downloaded from [45]. The embedding technique Steghide [46] is used to produce the stego images. We have selected Steghide over other embedding techniques such as F5 and Outguess as it is more secure according to [47]. A text file of size 1K is embedded into *jpeg* images. Only 3796 stego images are produced and their corresponding cover images are used for feature extraction. Steghide was unable to embed the other 1355 images because the images' size was not large enough to embed a 1K file. The data set of these extracted features is used in the training and the testing processes. A total of 169 features are extracted for each image. In the experiment, 50% of the data set is randomly selected for training process and 50% of the data set is used in the testing process. The mean absolute error for the trained network is 1.41%. The calculated testing accuracy is 99.01%. The calculated testing accuracy is based on the following equation.

$$Testing\ Accuracy = (TP + TN)/(TP + FP + TN + FN)$$

Where *TP*, *TN*, *FP* and *FN* are the number of true positive, true negative, false positive and false negative cases respectively.

We have used a neural network classifier which shows a 99.01% testing accuracy. This result outperforms an SVM classifier testing accuracy of 92.7% reported by Liu *et. al.* [39].

5.2 Improved Mel-Cepstrum Technique

A large data set of 16796 cover audio files and their corresponding stego files is used for feature extraction. Then, the extracted features are used in training and testing for each classifier to verify the performance of the improved Mel-Cepstrum technique. The files used in this section are randomly collected from the Internet. The following training and testing sets are used in our experiments:

1. A sample of 16782 mono cover *wav* audios and their corresponding 16782 mono stego *wav* audios with 100% hiding ratio to maximum embedding capacity produced by Steghide are used for feature extraction. The extracted features are used in the training and testing processes.
2. Steghide is used for embedding with 50% hiding ratio to the maximum embedding capacity. A sample of 16788 mono cover *wav* audios and their corresponding 16788 mono stego *wav* audios is used for features extraction. This data set of the extracted features is used in the training and testing processes.
3. Hide4PGP with 12.5 % hiding ratio to the maximum embedding capacity is used to produce the stego files. A sample of 16796 mono cover *wav* audios and their corresponding 16796 mono stego *wav* audios is used for features extraction. This data set of the extracted features is used in the training and testing processes.
4. Invisible Secrets 4 version 4.7 [48] with both 12.5 % and 25% hiding ratio to the maximum embedding capacity is used to produce the stego files. In each case, a sample of 16796 mono cover *wav* audios and their corresponding 16796 mono stego *wav* audios is used for features extraction. These data sets of the extracted features are used in the training and testing processes.

In these experiments, 50% of the audio files are randomly selected as a training set and the remaining 50% are used for testing. Feed-forward backpropagation neural networks based classifiers with three hidden layers are constructed and are used in the training and the testing processes. Also, SVMs classifiers are used for training and testing. Table 1 shows the results of six different classifiers used for training and testing.

The second column in the table shows the average testing accuracy results reported by Liu for a data set of 19380 using SVMs classifiers [37]. The data set used by Liu has similar characteristics (mono *wav* files 44.1 kHz with 16 bit PCM quantization. They are uncompressed and each file has duration of 10 seconds) as the data set we used in our experiments. The third column shows the testing accuracy results of the improved 2D Mel-Cepstrum using SVMs classifiers. The reported result for each embedding technique is the average of 10 experiments. The fourth column shows the testing accuracy results of AdaBoost technique that uses linear threshold classifiers available at [49]. A total of 18 linear threshold classifiers are used in the reported results. The training and testing processes are done for the extracted features from the cover files and their corresponding stego files that were embedded with Steghide using 50% hiding ratio to maximum embedding capacity. Then, the trained network is tested for the other embedding techniques as the reported results shown in the fourth column. The fifth, sixth, and seventh columns show the testing accuracy results for each embedding technique of the improved 2D Mel-Cepstrum using neural networks. The highest testing accuracy results are highlighted in bold. The fifth column shows the testing accuracy results for each embedding technique of the neural network *NN_Steg50* that is constructed and trained using data set obtained from the cover files and their corresponding stego files that were embedded with Steghide using 50% hiding ratio to maximum embedding capacity. The sixth column shows the testing accuracy results of the neural network *NN_I* that is constructed and trained using data set obtained from the cover files and their corresponding stego files that were embedded with the same embedding technique that is used for the testing set. For example, testing of the files that have embedding data generated by Invisible secrets is performed on a network trained on files that have embedding using Invisible secrets. For this case, the embedding ratios to maximum embedding capacity are 25% and 12.5% as shown in Table 1. The seventh column shows the testing accuracy results of the neural network *NN_Mix* that uses the data set of the extracted features that are generated from the cover files and their corresponding stego files that were embedded with a mix of Steghide, Invisible, and Hide4PGP. The stego mix has a total of 50380 files. These files include 16788 files embedded using Steghide with 50% embedding ratio to maximum embedding capacity, 16796 files embedded using Invisible Secrets with 12.5% embedding ratio to maximum embedding capacity, and 16796 files embedded using Hide4PGP with 12.5% embedding ratio to maximum embedding capacity.

By examining the results in Table 1, we have the following conclusions:

(1) All the three classification techniques for our improved 2-D Mel-Cepstrum implementation outperform the results reported by Liu [37].
(2) For the improved 2-D Mel-Cepstrum technique, the neural network classifiers outperform the SVMs classifiers and the AdaBoost classifiers.
(3) The difference in accuracy between neural networks classifiers, AdaBoost classifiers and SVMs classifiers for improved 2-D Mel-Cepstrum is within 3%.
(4) The best testing accuracy results obtained for the neural networks classifiers are those obtained from the network *NN_Mix* which uses a training set obtained by mixing the data sets from the three embedding techniques.

5.3 Extensible classifier to add a new embedding technique

Inspired by the methodology of AdaBoost, the following approach is presented to extend the current classifiers to allow adding new embedding technique(s) to the current supported embedding techniques. The following classifier combines different individual classifiers into a stronger classifier by using the boosting process. Note that the difference in this classifier versus the previously used AdaBoost classifier in this work, is that here we perform boosting based on the classifiers' performance using different embedding techniques, where as previously we utilized an AdaBoost classifier based on the boosting of 18 single feature linear threshold classifiers. In the extensible classifier, we use boosting on individually trained neural network classifiers for different embedding algorithms such as Invisible Secrets, Steghide, and Hide4PGP etc. The reason for choosing boosting of neural network (NN) based classifiers of individual embedding algorithms is because our NN classifiers performed better than SVMs and linear threshold AdaBoost classifiers. The overall structure of the extensible classifier is shown in Figure 2.

Definitions

(a) CL_i : A classifier designed for embedding technique ET_i
(b) Embedding Techniques Set (ETS) = (Steghide, Hide4PGP, Invisible Secrets, …)

Weight Set (WS) = (w_1, w_2, …, w_N) such that:

$$w_i = \log\left(\frac{1 - errorRate_i}{errorRate_i}\right) \quad \forall\ i = 1.2.\dots.N$$

$$errorRate_i = \frac{Number\ of\ misclassified\ datapoints\ for\ CL_i}{Total\ number\ of\ datapoints} \quad \forall\ i = 1.2.\dots.N$$

(c) Normalized Weight:

$$W_i = w_i / \textstyle\sum_{j=1}^{N} w_j \qquad \forall\, i = 1.2.\dots.N$$

Such that: $\sum_{i=1}^{N} W_i = 1$

(d) Output of CL_i (O_{CL_i}) = $\begin{cases} +1\ if\ data\ point\ \in class\ 1 \\ -1\ if\ data\ point\ \in class\ 2 \end{cases}$

Classification process

$$\text{Output of CL } (O_{CL}) = \begin{cases} +1\ if\ \sum_{i=1}^{N} W_i \times O_{CL_i} > 0 \\ -1\ if\ \sum_{i=1}^{N} W_i \times O_{CL_i} < 0 \end{cases}$$

Addition of a new classifier CL_{N+1}

Weight Set (WS) = (w_1, w_2, …, w_N, w_{N+1}) such that:

$$(a) w_i = \log\left(\frac{1 - errorRate_i}{errorRate_i}\right)\ \forall\ i = 1.2.\dots.N+1$$

$$(b)\ W_i = w_i / \textstyle\sum_{j=1}^{N+1} w_j \qquad \forall\, i = 1.2.\dots.N+1$$

Such that: $\sum_{i=1}^{N+1} W_i = 1$

The block diagram of the classifier obtained from the boosting of individual classifiers is shown in Figure 2. We present the results on this improved classifier in the next section.

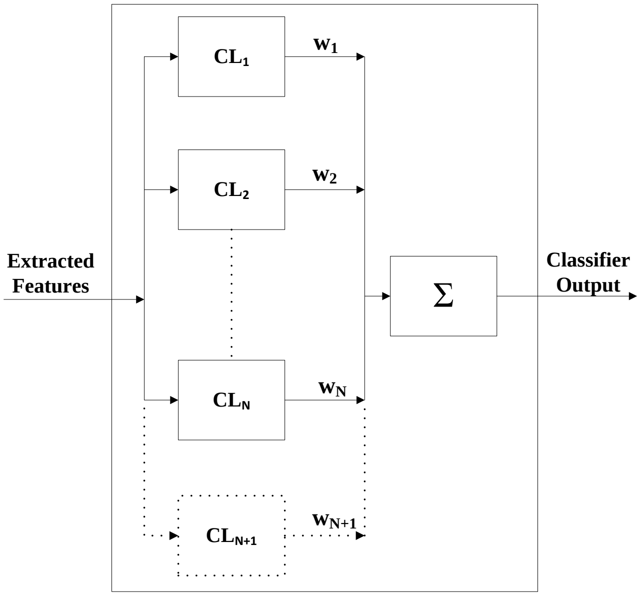


Fig 2: The overall structure of the extensible classifier

5.4 Enhancements to the initial framework for SOA based Steganalyzer

The test results presented in Section 5.2 have demonstrated that the three classification techniques for the improved 2-D Mel-Cepstrum outperform the results reported by Liu. Furthermore, experimental results of improved 2-D Mel-Cepstrum outperform the experimental results of Markov technique [37, 40]. Accordingly, we revised our initial architecture to remove the Markov service as it becomes non-beneficial to include it. Figure 3 shows the modified architecture where we include our new extensible classifier introduced in Section 5.3. Figure 4 shows the confusion matrix when using the extensible classifier. The testing accuracy obtained is 99.9% which is rounded to 100% as shown in the figure. This testing accuracy indicates that the developed extensible classifier outperforms all the testing accuracy obtained and reported earlier in this paper, as well as reported by other researchers. Thus, the extensible classifier not only allows adding new embedding technique(s), but also provides the best performance of all other classifiers.

6. CONCLUSIONS AND FUTURE WORK

We have developed an SOA based framework for steganalysis and implemented a unified, reliable Steganalyzer that supports multiple carrier types and embedding algorithms. Our implementation demonstrated two types of carrier objects: *wav* audios and *jpeg* images. In our implementation, reliability is achieved by integrating techniques that are accomplishing accurate results, and by the design of an efficient multi-stage neural network classifier for *jpeg* images, and an extensible classifier for *wav* audios. The developed architecture supports parallel processing which enhances the speed of the steganalyzer and is extensible to handle other data types that are not currently supported. We also developed an extensible classifier that allows addition of new embedding technique(s) so that the steganalyzer can reliably detect the stego object(s) that are generated by future embedding techniques. We have demonstrated that our system is capable of distinguishing between the cover and the stego objects with an average accuracy above 99%.

We have presented three different classifiers in this paper, and an extensible classifier that combines existing individual classifiers based on different embedding algorithms in a boosting process to enhance overall accuracy. The three classifiers implemented are Support Vector Machines, AdaBoost (based on the boosting of 18 single feature linear threshold classifiers) and Neural Networks. The results obtained from the three classifiers are within 3% of each other, whereas the Neural Networks had the best performance. Our extensible classifier combines individual classifiers based on different embedding techniques into a stronger overall classifier by using the boosting process. This classifier has led to a very high classification accuracy that is almost close to 100%. In the data set of 16796 cover files and their corresponding 16796 stego files, there are only 3 false negative results and 6 false positive results as shown in the confusion matrix in Figure 4. We also use the idea of AdaBoost based classifier as the basis for extensibility of our final classifier to allow adding new embedding technique(s), to continue to perform steganalysis reliably for future embedding algorithms.
There are some dimensions of research related to this paper that can be further investigated in the future e.g., integration of other media types such as gif, tiff, mp3, mpeg, and streaming media. Another issue that researchers face in the field of steganalysis is the unavailability of a benchmark that can be used to examine new and enhanced techniques for more rigorous and fair comparison. Also, it would be beneficial to test the possibility of adopting the Services Oriented Architecture framework presented in this paper in other classification problems such as face detection and recognition. This is particularly applicable in classification problems where multiple algorithms are involved. Finally, we would like to launch the services developed in this work as Internet services that can be accessed and used by the research community all over the world.

REFERENCES

[1] Fridrich, J.: Feature-based steganalysis for *jpeg* images and its implications for future design of steganographic schemes. Proc. 6th Information Hiding Workshop, Toronto. 67–81 (2004)

[2] Kharrazi, M., Sencar, H. T., Memon, N.: Benchmarking steganographic and steganalysis techniques. Security, Steganography, and Watermarking of Multimedia Contents VII. Edited by Delp, Edward J., III; Wong, Ping W. Proceedings of the SPIE, vol. 5681. 252-263 (2005)

[3] Martin, A., Sapiro, G., Seroussi, G.: Is image steganography natural? IEEE Transactions on Image Processing, issue 12. 2040 – 2050 December (2005)

[4] Upham, D.: JPEG-JSTEG—Modifications of the Independent JPEG Groups JPEG Software for 1-Bit Steganography in JFIF Output Files. ftp://ftp.funet.fi/pub/crypt/steganography/ Accessed 20 May 2011

[5] Tzschoppe, R., Bäuml, R., Huber, J. B., Kaup, A.: Steganographic system based on higher-order statistics. SPIE Security and Watermarking of Multimedia Contents V, Santa Clara, CA. (2003)

[6] Brown, A.: S-Tools for Windows, 1994. ftp://ftp.ntua.gr/pub/crypt/mirrors/idea.sec.dsi.unimi.it/code/s-tools4.zip Accessed 20 May 2011

[7] Gousseau, Y., Morel, J. M.: Are natural images of bounded variation. SIAM J. Math. Anal., vol. 33, no. 3. 634–648 (2001)

[8] Alvarez, L., Gousseau, Y., Morel, J. M.: The size of objects in natural images. CMLA. Cachan, France: Ecole Normale Sup. (1999)

[9] Grenander, U., Srivastava, A.: Probability models for clutter in natural images. IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 4. 424-429 Apr(2001)

[10] [Srivastava, A., Liu, X., Grenander, U.: Universal analytical forms for modeling image probabilities. IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 9. 1200 – 1214 (2002)

[11] Grenander, U.: Toward a theory of natural scenes. Technical report, Brown Univ., Providence, RI, (2003).

[12] Green, M. L.: Statistics of images, the TV algorithm of Rudin-Osher-Fatemi for image denoising and an improved denoising algorithm. Technical Report, Univ. California, Los Angeles, (2002).

[13] Avcibas, I., Memon, N., Sankur, B.: Steganalysis using image quality metrics. Proc. Security and Watermarking of Multimedia Contents, San Jose, CA. 523 – 531 (2001)

[14] Avcibas, I., Kharrazi, M., Memon, N., Sankur, B.: Image steganalysis with binary similarity measures. EURASIP J. Appl. Signal Process, vol. 17. 2749–2757 (2005)

[15] Farid, H.: Detecting hidden messages using higher-order statistical models. Proc. IEEE International Conference on Image Processing (ICIP '02), vol. 2, Rochester, NY, USA. 905–908 September (2002)

[16] Provos, N. Outguess. http://www.outguess.org/detection.php Accessed 20 May 2011

[17] Westfeld, A.: F5-A Steganographic Algorithm: high capacity despite better steganalysis. Proceedings of the 4th International Workshop on Information Hiding, Lecture Notes In Computer Science; vol. 2137. 289 - 302 (2001)

[18] Sallee, P.: Model-based steganography. Proc. Int. Workshop on Digital Watermarking, Seoul, Korea. 254-260 (2003)

[19] Fridrich, J., Goljan, M., Soukal, D.: Perturbed quantization steganography with wet paper codes. Proc. ACM Multimedia Workshop, Magdeburg, Germany. 4–15 (2004)

[20] Kharrazi, M., Sencar, H. T., Memon, N.: Performance study of common image steganography and steganalysis techniques. Journal of Electronic Imaging vol. 15, issue 4. Oct–Dec (2006)

[21] Lyu, S., Farid, H.: Detecting hidden messages using higher-order statistics and support vector machines. Proc. 5th Int. Workshop on Information Hiding. 340-354 (2002)

[22] Lyu, S., Farid, H.: Steganalysis using color wavelet statistics and one-class support vector machines. Proc. SPIE 5306. 35-45 (2004)

[23] Goljan, M., Fridrich, J., Holotyak, T.: New Blind Steganalysis and its Implications. Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, vol. 6072, San Jose. CA. 1-13 January (2006)

[24] Holotyak, T., Fridrich, Voloshynovskiy, J. S.: Blind Statistical Steganalysis of Additive Steganography Using Wavelet Higher Order Statistics. 9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, LNCS vol. 3677, Springer-Verlag, Berlin. 273–274 (2005)

[25] Tian, H., Zhou, K., Jiang, H., Liu, J., Huang, Y., Feng, D.: An M-Sequence Based Steganography Model for Voice over IP. ICC '09. IEEE International Conference on Communications. 1-5 August (2009)

[26] Liu, Q., Sung, A. H., Qiao, M.: Temporal Derivative-Based Spectrum and Mel-Cepstrum Audio Steganalysis. IEEE Transactions on Information Forensics and Security, vol. 4, no. 3. 359-368 September (2009)

[27] Tian, H., Zhou, K., Jiang, H., Liu, J., Huang, Y., Feng, D.: An adaptive Steganography Scheme for Voice over IP. The 2009 IEEE International Symposium on Circuits and Systems. 2922-2925 (2009)

[28] Liu, Q., Sung, A. H., Qiao, M.: Novel Stream Mining for Audio Steganalysis. Proceedings of the seventeen ACM international conference on Multimedia, Beijing, China. 95 – 104 October (2009)

[29] Qiao, M., Sung, A. H., Liu, Q.: Feature Mining and Intelligent Computing for MP3 Steganalysis. 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing. 627-630 (2009)

[30] Introduction to Degol. http://wandership.ca/projects/deogol/intro.html Accessed 20 May 2011

[31] Li, Z., Sun, X., Wang, B., Wang, X.: A Steganography Scheme in P2P Network. IIHMSP '08 International Conference on Intelligent Information Hiding and Multimedia Signal Processing. 20 – 24 Aug (2008)

[32] Lalbakhsh, P., Ravanbakhsh, S., Fesharaki, M. N., Sohrabi, N.: Service Oriented Steganography - A novel approach towards autonomic secured distributed heterogeneous environments. 2009 International Conference on Signal Processing Systems, Singapore. 418 - 422 (2009)

[33] Papazoglou, M. P., Heuvel, W.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal — The International Journal on Very Large Data Bases, Springer-Verlag New York, Inc., vol. 16, issue 3. 389 - 415 July (2007)

[34] MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., Metz, R.: OASIS Reference Model for Service Oriented Architecture 1.0. Organization for the Advancement of Structured Information Standards (OASIS), Committee Specification 1. August (2006) http://www.oasis-

open.org/committees/download.php/19679/soa-rm-cs.pdf Accessed 20 May 2011

[35] Sprott, D., Wilkes, L. Understanding Service-Oriented Architecture. http://msdn.microsoft.com/en-us/library/aa480021.aspx Accessed 20 May 2011

[36] He, H. What Is Service-Oriented Architecture. http://www.xml.com/pub/a/ws/2003/09/30/soa.html Accessed 20 May 2011

[37] Liu, Q., Sung, A. H., Qiao, M.: Derivative Based Audio Steganalysis. ACM Transactions on Multimedia Computing, Communications and Applications, in press.

[38] Gonzalez, R., Woods, R.: Digital Image Processing. 3rd edition, Prentice Hall (2008)

[39] Liu, Q., Sung, A. H., Qiao, M.: Improved Detection and Evaluation for JPEG Steganalysis. Proceedings of the seventeen ACM international conference on Multimedia, Beijing, China. 873-876 October (2009)

[40] Abdelfattah, E., Mahmood, A. A Multi-Algorithm, High Reliability Steganalyzer based on Services Oriented Architecture. International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering December (2010)

[41] Ellis, D.: PLP and RASTA (and MFCC, and inversion) in Matlab using melfcc.m and invmelfcc.m. http://labrosa.ee.columbia.edu/matlab/rastamat/ Accessed 20 May 2011

[42] Freund, Y., Shapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Proceedings of the Second European Conference on Computational Learning Theory. 23 – 37 (1995)

[43] Rojas, R.: AdaBoost and the Super Bowl of Classifiers A Tutorial Introduction to Adaptive Boosting. (2009)

[44] Zilouchian, A., Jamshidi, M.: Intelligent Control Systems Using Soft Computing Methodologies. Chapter 2, Fundamentals of Neural Networks, CRC Press. (2001)

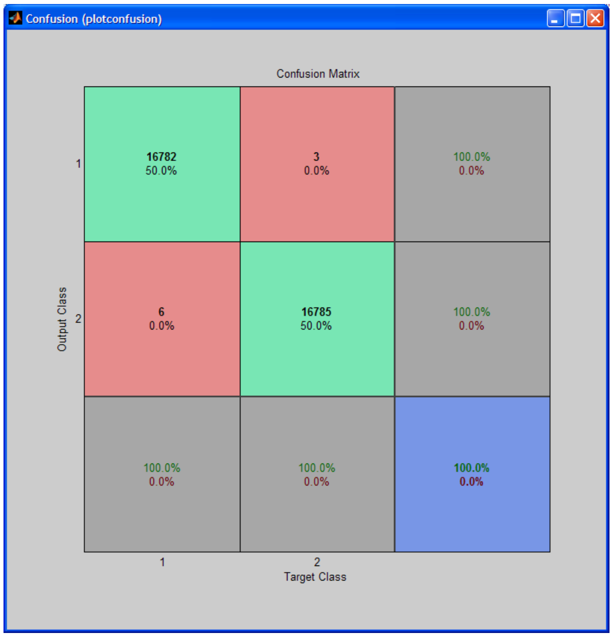
[45] Liu, Q. http://www.cs.nmt.edu/~liu/downloads.html Accessed 20 May 2011

[46] Steghide. http://steghide.sourceforge.net/ Accessed 20 May 2011

[47] Quach,T.: Information Similarity Metrics in Information Security and Forensics. Ph.D. Dissertation, University of New Mexico, Albuquerque. (2009)

[48] Invisible Secrets 4. http://www.invisiblesecrets.com/ Accessed 20 May 2011

[49] Mertayak, C. AdaBoost. http://www.mathworks.com/matlabcentral/fileexchange/21317-adaboost Accessed 20 May 2011



Confusion (plotconfusion)

Confusion Matrix

Output Class

	1	2	
1	16782 50.0%	3 0.0%	100.0% 0.0%
2	6 0.0%	16785 50.0%	100.0% 0.0%
	100.0% 0.0%	100.0% 0.0%	100.0% 0.0%

Target Class

Fig 4: Confusion matrix when using the extensible classifier

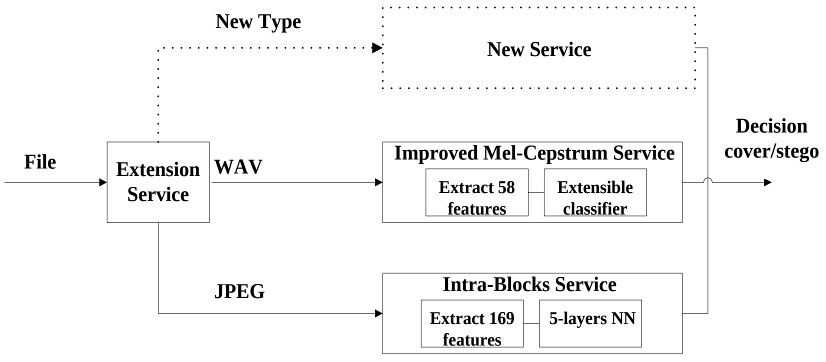


Fig 3: A modified architecture with the extensible classifier

Table 1. Testing Accuracy for Improved Mel-Cepstrum using Large Data Set

Extraction Technique Embedding Method	**2D-Mel**	**Improved 2D-Mel**				
		SVMs	**AdaBoost**	**NN_Steg50**	**NN_I**	**NN_Mix**
Steghide	100% hiding					
	95.8	96.39	97.82	98.2	99.0	**98.7**
	50 % hiding					
	92.7	95.61	97.41	97.4	97.4	**98.2**
Invisible	100% hiding	25% hiding				
	96.6	98.32	98.83	99.8	**100**	99.8
	50% hiding	12.5% hiding				
	89.9	98.32	98.83	99.8	**100**	99.8
Hide4PGP	25% hiding	12.5% hiding				
	96.7	96.76	98.93	99.6	99	**99.7**